CLAIMS

What is claimed is:

1. A system to facilitate generalized comprehension in an imperative language, comprising:

a language component to enable programming of comprehension notations in an imperative language;

an interface component to describe a meaning of the comprehension notations; and

a translation component to facilitate execution of the comprehension notations in accordance with the imperative language.

2. The system of claim 1, the language component includes a generalized comprehension that performs operations on a fixed or list comprehension.

3. The system of claim 1, the interface component defines one or more methods for the generalized comprehension.

4. The system of claim 1, the translation component includes at least one of just-in-time compilation techniques, interpretive techniques, and source code compilation techniques.

5. The system of claim 2, the language component enables users to define at least one of an implicit expression, an explicit expression, a mathematical expression, a database expression, and a processing expression in accordance with the generalized comprehension.

6. The system of claim 1, the comprehension notations include at least one of the following syntax:

comprehension ::= type{ expression : qualifiers }
qualifiers ::= qualifier (, qualifier)*
qualifier ::= generator | filter | local declaration
generator ::= type$_{opt}$ identifier in expression
filter ::= expression.

7. The system of claim 1, the interface component is associated with at least one of an IBuildable interface and an IBuilder interface.

8. The system of claim 7, the interface component is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

9. The system of claim 7, further comprising an assignment expression or a yield return statement.

10. The system of claim 7, further comprising an IEnumerable or IEnumerator component.

11. The system of claim 7, further comprising a relational database expression.

12. The system of claim 11, the relational database expression is employed for a query of a database.

13. A computer readable medium having computer readable instructions stored thereon for implementing the language component, the interface component, and the framework component of claim 1.

14. A comprehension notation system, comprising:

means for defining a list comprehension set and a generalized comprehension expression, the generalized comprehension expression defined exterior to the list comprehension;

means for associating the list comprehension set with the generalized comprehension expression; and

means for providing an interface for the generalized comprehension expression.

15. The system of claim 14, further comprising means for compiling the list comprehension, the generalized comprehension expression, and the interface.

16. A method for providing generalized comprehension in an imperative language, comprising:

defining a list comprehension expression; and

defining a generalized comprehension class as an exterior component to the list comprehension expression within an imperative language environment.

17. The method of claim 16, further comprising providing an interface class for the generalized comprehension class.

18. The method of claim 16, further comprising defining a results function for the interface class.

19. The method of claim 18, the results function returns a type that is at least one of similar and dissimilar to a type associated with the generalized comprehension class.

20. The method of claim 16, further comprising compiling the list comprehension expression and the generalized comprehension class to produce an executable format for the imperative language environment.

21. The method of claim 16, further comprising defining at least one relational database expression.

22. A computer readable medium having a data structure store thereon, comprising:

a first data field to define a static comprehension notation;

a second data field to define a generalized comprehension notation; and

a third data field to link the static comprehension notation with the generalized comprehension notation.

23. The medium of claim 22, further comprising an interface field and at least one method associated with the interface field.

24. The medium of claim 22, the interface field is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

25. The medium of claim 22, the generalized comprehension notation is associated with a user-defined expression.

26. The medium of claim 22, the generalized comprehension notation is associated with at least one of a Sum, Average, Product, Forall, Exists, Choose, Max, Seq, Multiset, Array, Add, and Query expression.

27. The medium of claim 26, further comprising a semantically equivalent expression.

28. The medium of claim 22, further comprising a comprehension type for a direct aggregation of collections.

29. The medium of claim 22, further comprising a field for an evaluation that is deduced from analyzing a portion of a collection.

30. The medium of claim 22, further comprising a field associated with at least one of a default value and an initialization value.

31. The medium of claim 22, further comprising a comprehension type that implicitly implements an interface.

32. The medium of claim 22, further comprising an interface pattern for defining aggregation functions on collections.